



RESPONSE UNDER 37 C.F.R. §1.116
EXPEDITED PROCEDURE - EXAMINING GROUP [2100]

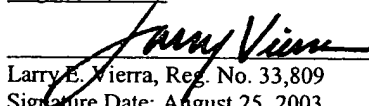
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED
JAN 08 2004
OFFICE OF PETITIONS

In re Application)	<u>PATENT APPLICATION</u>
Inventors: David L. Multer, et al.)	
Appl. No.: 09/642,615)	Art Unit: 2177
Filed: August 17, 2000)	Examiner: Channavajjala, S.
Title: CHANGE LOG AGGREGATION AND OPTIMIZATION)	<u>Customer No. 28554</u>

CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited in the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on August 25, 2003.


Larry E. Vierra, Reg. No. 33,809
Signature Date: August 25, 2003

RESPONSE B TO OFFICE ACTION UNDER 37 C.F.R. § 1.116

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This RESPONSE B is in reply to the Office Action mailed April 25, 2003.

AMENDMENTS to the CLAIMS begin on Page 2 of this paper.

REMARKS begin on Page 6 of this paper.

Amendments to the Claims

This listing of claim will replace all prior versions and listings of claim in the application.

1. (Currently amended) A method for updating application data in a client device of a data transfer and synchronization system, said method comprising the steps of:

downloading a first change log of a plurality of change logs from a server system, each of said plurality of change logs reflecting changes to said application data;

adding said first change log to an aggregate change log, the aggregate change log comprising a summary of changes in said added change log and any previous change logs;

deleting said first change log;

repeating said downloading, adding, and deleting steps for a next change log of said plurality of change logs until no additional change logs exist; and

applying said aggregate change log to said application data to update said application data.

2. (Original) The method of claim 1, wherein said adding step further comprises the steps of:

- (a) retrieving information for a valid item in said application data;
- (b) updating a map of said aggregate log, said map storing meta-data;
- (c) writing said item to said aggregate log;
- (d) updating a location of said valid item in said map; and
- (e) repeating steps (a) – (d) for all remaining valid items of a current change log.

3. (Original) The method of claim 2, further comprising the step of:
compacting said aggregate log if a compact threshold is exceeded.

4. (Original) The method of claim 1, wherein said application data comprises data classes for contacts, internet browser bookmarks, calendar events, email messages, notes, tasks, and files.

5. (Original) The method of claim 4, wherein said contacts comprise records identifying names, addresses, phone numbers, and email addresses for a plurality of individuals.

6. (Original) The method of claim 4, wherein said files comprise word processor specific documents, electronic presentations, spreadsheets, and executable files in binary format.

7. (Original) The method of claim 1, wherein said application data is in a universal data format.

8. (Original) An apparatus for updating application data in a client device of a data transfer and synchronization system, said apparatus comprising:

a downloading routine for iteratively retrieving a plurality of change logs from a server system, each of said plurality of change logs reflecting changes to said application data;

a merging routine for iteratively aggregating the contents of said plurality of change logs to an aggregate log;

a change log deletion routine for iteratively deleting said plurality of change logs; and

an updating routine for applying the contents of said aggregate log to said application data to update said application data.

9. (Original) The apparatus of claim 8, wherein said merging routine further comprises:

an item retrieval routine for retrieving a plurality of items from said application data;

a map update routine for updating a map of said aggregate log wherein said map stores meta-data;

a field retrieval routine for retrieving field information from said plurality of items from said application data; and

an item location update routing for updating the location of a plurality of items in said map.

10. (Original) The apparatus of claim 9, wherein said merging routine further comprises:

an aggregate log compacting routine.

11. (Currently Amended) A method for aggregating the contents of accumulated change logs into an aggregate log and applying said aggregate log to update application data in a first client device of a data transfer and synchronization system, said method comprising the steps of:

downloading to the first client device a first change log of a plurality of change logs [from] stored on a server system, each of said plurality of change logs reflecting [changes] transactions to said application data;

adding said first change log to an aggregate log on the first client device;

deleting said first change log;

repeating said downloading, adding, and deleting steps for a next change log of said plurality of change logs until no additional change logs exist; and

applying said aggregate log to said application data to update said application data.

12. (Previously Added) The method of claim 11 wherein the log includes an identifier for a change.

13. (Previously Added) The method of claim 12 wherein the step of applying said aggregate log includes requesting a binary delta for a file.

cl 14. (Previously Added) The method of claim 11 further including the step of creating the aggregate log.

15. (Previously Added) The method of claim 11 further including the step of closing the aggregate log.

16. (Previously Added) The method of claim 11 further including the step of compacting said aggregate log.

17. (Previously Added) The method of claim 16 wherein said step of compacting comprises:

iterating over each record in the aggregate log;

reading valid records; and

writing valid records back to the aggregate log overwriting obsolete records.

18. (New) A method of creating a composite change log on a client device, comprising:
downloading a first change log of a plurality of change logs from a server system, each of
said plurality of change logs reflecting changes to said application data;

cl adding said first change log to an aggregate change log, the aggregate change log comprising
a summary of changes in said added change log and any previous change logs;

deleting said first change log; and

repeating said downloading, adding, and deleting steps for a next change log of said plurality
of change logs until no additional change logs exist.

Remarks

The above Amendments and these Remarks are in reply to the Office action mailed April 25, 2003. Claims 1 – 18 are presented herewith for consideration.

Summary of the Examiners Rejections

Claims 1-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Birkler, in view of Bowen et al.

Summary of the Amendments

Claims 1 and 11 have been amended; claim 18 has been added.

Remarks

Entry of the Amendment

Applicant submits there are good and sufficient reasons why such amendments were not earlier submitted, namely that the amendments advance the prosecution of the application and place the application in condition for allowance. Entry of the Amendments is therefore respectfully requested.

Co-Pending Applications

The Examiner has requested that the status of the co-pending applications cited on Page 8 of the application be updated. The applications are currently in prosecution and there are no additional changes to the serial numbers cited in the application. Application serial no. 09/491,694 has been allowed, but no Issue Notification has yet been received.

The Invention Defined in the Claims is Not Obvious

It is respectfully submitted that the claimed invention is not obvious over Birkler in view of Bowen et al.

Neither reference teaches a fundamental limitation of the present invention, that of an "aggregate change log". In both references, the techniques described update a database or

“application data”, not a change log or “aggregate change log”, which is then used to update application data. Hence, a fundamental feature of the invention present in all the claims is not taught by the references.

The Examiner recognizes that Birkler does not teach “aggregate log, applying said aggregate log to said application data to update said application data.” (Office Action, page 5, line 9 et seq.) To find this feature, the Examiner turns to the Bowen et al. reference. However, a fair and detailed reading of the Bowen et al. reference reveals that the reference does not describe this feature.

As discussed below, in Bowen et al. a computer 20 executing a read request is directed to an aggregation system 30. The aggregation system maintains the integrity of the data by holding a base value and aggregating that value with incremental change values in response to a read request. (Col 4, lines 14 - 23). Hence, the aggregation system is, in effect, the data store. The data is contained in two “logs” or data structures – a base structure and an incremental structure. (id.) Periodically, the base structure is updated by the incremental structure, but the combination of the base and incremental structures provides the data which answers the “read” request. (Col. 7, lines 44 – 47, lines 19 – 44, respectively).

In particular, in Bowen et al.:

1. It is the base log, or the base log plus the incremental log values, which are read by the computer. Hence, the “aggregation” is the combination of the incremental log and the base log to provide the read request response value. There is no creation of, or suggestion to create, an “aggregate change log.”
2. The Base Log is updated by the Incremental Log, but the base log is the data which is read. Neither the base log nor the incremental log is used to update another database.

In addition, contrary to the Examiner’s assertion, Birkler does not teach optimizing a “change log”, but rather optimizing *the use* of a change log in a synchronization process. The log itself does not change.

1. There Is No Creation Of, Or Suggestion To Create, An Aggregate Change Log.

Bowen et al. does not teach an “aggregate change log”. The aggregation system, while aggregating values, does not provide an aggregate log.

In Bowen et al.:

an update of a data value in a database is transformed from an update operation that changes a unique value in the database to a logging operation that records incremental changes in a data value. A database read operation is correspondingly transformed from a read of a single value, to a selection of a base value and the aggregation of all the incremental changes in the log associated with the selected base value. (Col. 4, lines 17 – 24).

Hence, the base log, or the base log plus the incremental log values, are read by the general purpose computer (e.g. provided by the aggregation system in response to the read request). The base and incremental logs are described with respect to Fig. 4:

FIG. 4 schematically illustrates the *log relations stored in the memory 32*. This relation includes three attributes: item identification (i), delta value (δ_{ij}) and time-stamp (t_{ij}). (Col. 6, lines 33 – 35, emphasis supplied).

These logs describe tables in the database, as clearly indicated by the specification:

a relational database, which is also known as a **relation**, data is organized in columns. Each column comprises one attribute of the relation. Each column or attribute of a relation has a domain which comprises the data values in that column. (Col. 1, lines 44 - 48).

Hence neither of these logs is an “aggregate change log”, as required by

Read requests are provided by the host computer and answered by the aggregation system.

To read a value, Bowen et al. provides that the system:

1. select the base value $base_i$ and the time-stamp t_i from the base relation;
2. select from the log relation the increments δ_{ij} having a time-stamp t_{ij} which is later than t_i ; and
3. compute $N_i = base_i + \sum \delta_{ij}$. (Col. 6, lines 37 – 42).

Bowen et al. makes clear that the “base log” is equivalent to a “database” in the claim, while a “incremental log” is equivalent to a “change log”. There is no “aggregate change log”:

The incremental updates represent incremental changes in the data values. Therefore, to compute the current value of a numeric summary value, incremental updates associated with the numeric summary value and having a time stamp later

than the time-stamp of the base value associated with the numeric summary value are aggregated with the base value. (Col. 5, lines 53-59).

Hence, "aggregation" in the context of the Bowen et al. reference does not mean generation of an aggregate change log. Incremental values are shown in Figure 4. The "aggregation" is the combination of the incremental log and the base log to provide the read request response value.

As noted above, the aggregation system's job is to return the read request value to the computer. It does this by adding the base and incremental value.

The Examiner states that:

One of ordinary skill in the art at the time of the inventing would have been motivated to modify the Birkler's reference... to incorporate the aggregation system of ... Bowen et al. because that would have allowed users of Birkler optimization of synchronization procedures that utilize a change log system to control the aggregation system processor to update and maintain in the log..." (Office Action of 4/25/03, Page 6 lines 5 - 13).

However, Birkler teaches how to apply the log – via a full (slow) sync, a partial slow sync or a fast sync, depending on the knowledge both systems have about the change log. Bowen et al. teaches a processor to apply a specific log; it does not teach or suggest providing an aggregate log, or adding change logs to its change base log, to achieve an aggregate log.

2. The base log is, at most, the "application data" which is to be updated, not an "aggregate change log."

Neither the base log nor the incremental log is used to update another database. However...

... periodically, the base values in the base relation are updated and the incremental updates incorporated into the updated base values are removed from the log relation. Col. 7, lines 19 – 47.

Periodically, the incremental values are added to the base relation i.e. the database is updated. At most, this is equivalent to adding a change log to "application data". There is no aggregation of the change log (or incremental log) itself, but rather a "clearing" of the incremental data and an update of the base relation.

Hence, modifying Birkler with Bowen et al. would teach that in Birkler, one would provide

the updated data in a streaming form via an aggregation processor, and download that information and any incremental changes on a read request by the system seeking to be updated in Bowen et al. However, this is not equivalent to providing an aggregate change log, and applying that change log to the system's application data.

One of average skill in the art would not be led by the teachings of Birkler and Bowen et al. to the invention defined in claim 1 as the limitation calling for:

adding said first change log to an aggregate change log, the aggregate change log comprising a summary of changes in said added change log and any previous change logs;

Likewise one of average skill in the art would not be led by the teachings of Birkler and Bowen et al. to the invention defined in claim 8 calling for:

a merging routine for iteratively aggregating the contents of said plurality of change logs to an aggregate log;

Since there is no teaching of an aggregate log, there can be no "merging routine" as defined in claim 8.

Further, one of average skill in the art would not be led by the teachings of Birkler and Bowen et al. to the invention defined in claim 17 calling for:

adding said first change log to an aggregate change log, the aggregate change log comprising a summary of changes in said added change log and any previous change logs;

Again, there is no teaching of a step of "adding" since there is no "aggregate change log" as required by the limitation.

The portions of Birkler and Bowen et al. cited by the Examiner as allegedly providing the teachings of an aggregate log do not provide such teachings. The Examiner cites the passages at Col. 4, lines 11 - 14 and 45 - 50 of Birkler as teaching "adding... deleting" (Page 5 lines 7 of the Office Action dated 4/25/03) as teaching this feature.

Col 4, line 11 - 15 provides:

The present invention relates to a system and method for maintaining the consistency of data values such as aggregate numeric data values while allowing concurrent updates without the use of locking operations.

Col. 4, lines 45 – 50 provide:

The aggregation system also comprises a processor. The processor is in communication with the general purpose computer. The aggregation system processor performs update operations by receiving the incremental updates from the general purpose computer and writing them into the log relation maintained in the memory means.

Neither of these passages provides an aggregate change log. In fact, this latter passage is from the Summary of the Invention, and clearly, when read in light of other portions of the specification, describes no more than updating the base relation with the incremental changes, which would be equivalent to updating the database or, in the language of claim 1, applying a single change log to the database which needs to be updated.

The Examiner also cites Col. 4, lines 15 – 27 provide:

...A database read operation is correspondingly transformed from a read of a single value, to a selection of a base value and the aggregation of all the incremental changes in the log associated with the selected base value. Since logging operations avoid the locking of the database records for extended periods of time, processing serialization delays caused by lock contention during updates are avoided and greater levels of concurrency are achieved.

In other words, to obtain a value from the base log database, one applies the changes from the incremental log and sums it with those in the base log. Again, no teaching of an aggregate log is provided.

Based on the above amendments and these remarks, reconsideration of claims 1 - 17 and consideration of claim 18 is respectfully requested.


The Examiner's prompt attention to this matter is greatly appreciated. Should further questions remain, the Examiner is invited to contact the undersigned attorney by telephone.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 501826 for any matter in connection with this response, including any fee for extension of time, which may be required.

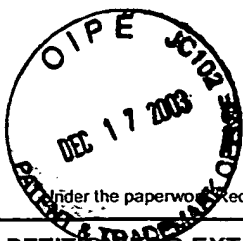
Respectfully submitted,

Date: August 25, 2003

By: _____


Larry E. Vierra
Reg. No. 33,809

VIERRA MAGEN MARCUS HARMON & DENIRO LLP
685 Market Street, Suite 540
San Francisco, CA 94105-4206
Telephone: (415) 369-9660
Facsimile: (415) 369-9665



PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)		Docket Number (Optional) FUSN1-01017US0	
In re Application of		Multer	
Application Number	09/642,615	Filed	8/17/2000
For		Change Log Aggregation and Optimization	
Art Unit	2177	Examiner	Channavajjala, Srirama T.

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and appropriate non-small-entity fee are as follows (check time period desired):

<input checked="" type="checkbox"/> One month (37 CFR 1.17(a)(1))	\$ 110.00
<input type="checkbox"/> Two months (37 CFR 1.17(a)(2))	\$
<input type="checkbox"/> Three months (37 CFR 1.17(a)(3))	\$
<input type="checkbox"/> Four months (37 CFR 1.17(a)(4))	\$
<input type="checkbox"/> Five months (37 CFR 1.17(a)(5))	\$

☒ Applicant claims small entity status. See 37 CFR 1.27. Therefore, the fee amount shown above is reduced by one-half, and the resulting fee is: \$ 55.00

☒ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 501826

I have enclosed a duplicate copy of this sheet.

I am the ☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed (Form PTO/SB/96).

☒ attorney or agent of record. Registration Number 33,809

☐ attorney or agent under 37 CFR 1.34(a).
Registration number if acting under 37 CFR 1.34(a)

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

<u>8/25/03</u> Date	 Signature
<u>415-369-9660</u> Telephone Number	<u>Larry E. Vierra</u> Typed or printed name

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☐ Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.136(a). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 6 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.